SYSTEM AND METHOD FOR CHANGING HLD SPECIFICATION AND STORAGE MEDIUM RECORDING ITS CONTROL PROGRAM

Patent Number: JP2000148804
Publication date: 2000-05-30
Inventor(s): IIZUKA TAKUYA
Applicant(s):: NEC ENG LTD
Requested Patent: JP2000148804 (JP00148804)
Application Number: JP19980315723 19981106
Priority Number(s):
IPC Classification: G06F17/50
EC Classification:
Equivalents:

Abstract

PROBLEM TO BE SOLVED: To provide an HDL(hardware description language) specification changing system which can easily change various kinds of HDLs to each other without using hands nor preparing programs at every kind of HDLs.
SOLUTION: A syntactic element fetching means 21 calls the HDL template matching the kind of an HDL read from an HDL input device 1 from an HDL template storing section 31, fetches each syntactic element by using the template, and stores the elements in a syntactic element storing section 32. An HDL composing means 22 calls the HDL template matching the kind of an HDL to be changed from the storing section 31 and recomposes an HDL by combining the reserved word of the HDL to be changed with the syntactic element by referring to the HDL template and an HDL output device 4 outputs the recomposed HDL to a display device or another storing medium.

(d)

(19)日本国特許庁(JP)　(12)公開特許公報（A）　(11)特許出願公開番号

特開２０００－１４８８０４
（Ｐ２０００－１４８８０４Ａ）
(43)公開日　平成12年5月30日(2000.5.30)

(51)Int.Cl.⁷	識別記号	ＦＩ			テーマコード(参考)
Ｇ０６Ｆ　17/50		Ｇ０６Ｆ	15/60	６５４　Ａ	5B046

審査請求　未請求　請求項の数５　ＯＬ　(全７頁)

(21)出願番号　特願平10-315723

(22)出願日　平成10年11月6日(1998.11.6)

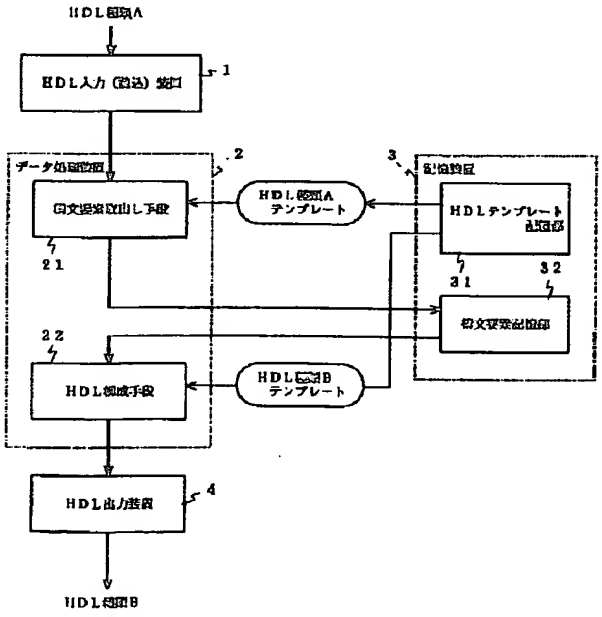
請求項3.

(71)出願人　000232047
日本電気エンジニアリング株式会社
東京都港区芝浦三丁目18番21号
(72)発明者　飯塚　卓也
東京都港区芝浦三丁目18番21号　日本電気エンジニアリング株式会社内
(74)代理人　100082935
弁理士　京本　直樹　(外2名)

Ｆターム(参考)　5B046 AA08 BA03 GA01 KA06

(54)【発明の名称】ＨＤＬ仕様変換システム及びそのＨＤＬ仕様変換方法並びにその制御プログラムを記録した記録媒体

(57)【要約】
【課題】　人手を使わず、またＨＤＬ種類個別にプログラムを用意することなく、容易に多種のＨＤＬ同士を変換可能なＨＤＬ仕様変換システムを提供する。
【解決手段】　構文要素取出し手段２１はＨＤＬ入力装置１から読込まれたＨＤＬの種類に合ったＨＤＬテンプレートをＨＤＬテンプレート記憶部３１から呼出し、そのＨＤＬテンプレートを使用してＨＤＬから各構文要素を取出し、構成要素記憶部３２に格納される。ＨＤＬ構成手段２２は変換したいＨＤＬの種類に合ったＨＤＬテンプレートをＨＤＬテンプレート記憶部３１から呼出し、そのＨＤＬテンプレートを参照して変換したいＨＤＬの予約語と構文要素とを合わせてＨＤＬに再構成し、ＨＤＬ出力装置４によってディスプレイ装置や他の記憶媒体に出力される。



1

【特許請求の範囲】

【請求項１】　複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたＨＤＬテンプレートを記憶するテンプレートライブラリと、入力されたハードウェア記述言語から前記テンプレートライブラリを参照して前記構文要素を取出す構文要素取出し手段と、前記構文要素取出し手段で取出された構文要素を格納する格納手段と、前記格納手段に格納された前記構文要素を基に変換したいハードウェア記述言語に対応するＨＤＬテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換する変換手段とを有することを特徴とするＨＤＬ仕様変換システム。

【請求項２】　前記変換手段は、前記格納手段に格納された前記構文要素と前記ＨＤＬテンプレートの予約語とを合わせて前記変換したいハードウェア記述言語に再構成するよう構成したことを請求項１記載のＨＤＬ仕様変換システム。

【請求項３】　複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたＨＤＬテンプレートを記憶するテンプレートライブラリを参照して入力されたハードウェア記述言語から前記構文要素を取出すステップと、その取出された構文要素を格納するステップと、格納された前記構文要素を基に変換したいハードウェア記述言語に対応する前記ＨＤＬテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換するステップとを有することを特徴とするＨＤＬ仕様変換方法。

【請求項４】　前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換するステップは、格納された前記構文要素と前記ＨＤＬテンプレートの予約語とを合わせて前記変換したいハードウェア記述言語に再構成するようにしたことを請求項３記載のＨＤＬ仕様変換方法。

【請求項５】　コンピュータに、入力されたハードウェア記述言語を変換したいハードウェア記述言語に変換させるためのＨＤＬ仕様変換制御プログラムを記録した記録媒体であって、前記ＨＤＬ仕様変換制御プログラムは前記コンピュータに、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたＨＤＬテンプレートを記憶するテンプレートライブラリを参照して前記入力されたハードウェア記述言語から前記構文要素を取出させ、その取出させた構文要素を格納させ、格納させた前記構文要素を基に前記変換したいハードウェア記述言語に対応する前記ＨＤＬテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換させることを特徴とするＨＤＬ仕様変換制御プログラムを記録した記録媒体。

2

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はＨＤＬ仕様変換システム及びそのＨＤＬ仕様変換方法並びにその制御プログラムを記録した記録媒体に関し、特にＬＳＩ（大規模集積回路）における論理設計の支援をする支援装置に関する。

【０００２】

【従来の技術】従来、ＬＳＩ設計に使用するＨＤＬ［Ｈａｒｄｗａｒｅ　Ｄｅｓｃｒｉｐｔｉｏｎ　Ｌａｎｇｕａｇｅｓ：ハードウェア記述言語（様々な電子回路の構造及び動作を表現するための言語）］には様々な種類が存在する。しかしながら、ＨＤＬを入力とするＬＳＩ設計支援プログラムが、全ての種類のＨＤＬに対応している訳ではない。

【０００３】ＨＤＬに合わせたプログラムを使用する必要があるが、必ずしもそのようなプログラムが存在するとは限らない。そこで、ＨＤＬの種類の変換が必要になる。しかも、ＨＤＬの内容の可読性、テクノロジの汎用性等も考慮すると、ＲＴレベル記述［Ｒｅｇｉｓｔｅｒ　Ｔｒａｎｓｆｅｒ　Ｌｅｖｅｌ：レジスタ、カウンタ、マルチプレクサ、ＡＬＵ（Ａｒｉｔｈｍｅｔｉｃ　ａｎｄ　Ｌｏｇｉｃ　Ｕｎｉｔ）等のレベルで機能を表現する記述］でのＨＤＬの種類変換が必要となる。

【０００４】

【発明が解決しようとする課題】上述した従来の異なる種類のＨＤＬ間の変換方法では、人手によって新たに作成し直すか、あるいは特定される１種類対１種類の変換を直接行うプログラムを実行するしかない。

【０００５】また、間接的な方法で実際に論理合成やＨＤＬシミュレーション時に種類に合わせたコンパイラを使用する方法があるが、この方法の場合、ＨＤＬ種類に合うコンパイラがあるとは限らず、またコンパイラを通すとＲＴレベル記述で出力できないゲートレベル［ＲＴレベルよりさらに抽象性の低いレベルであり、アンド（ＡＮＤ）、オア（ＯＲ）、インバータ（ＩＮＶＥＲＴ）の他、幾つかのフリップ・フロップで機能を表現したもの）になってしまう等の制約がある。

【０００６】そこで、本発明の目的は上記の問題点を解消し、人手を使わず、またＨＤＬ種類個別にプログラムを用意することなく、容易に多種のＨＤＬ同士を変換することができるＨＤＬ仕様変換システム及びそのＨＤＬ仕様変換方法並びにその制御プログラムを記録した記録媒体を提供することにある。

【０００７】

【課題を解決するための手段】本発明によるＨＤＬ仕様変換システムは、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたＨＤＬテンプレートを記憶するテンプレートライブラリと、入力されたハードウェア記述言語から前記

テンプレートライブラリを参照して前記構文要素を取出す構文要素取出し手段と、前記構文要素取出し手段で取出された構文要素を格納する格納手段と、前記格納手段に格納された前記構文要素を基に変換したいハードウェア記述言語に対応するＨＤＬテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換する変換手段とを備えている。

【０００８】本発明によるＨＤＬ仕様変換方法は、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたＨＤＬテンプレートを記憶するテンプレートライブラリを参照して入力されたハードウェア記述言語から前記構文要素を取出すステップと、その取出された構文要素を格納するステップと、格納された前記構文要素を基に変換したいハードウェア記述言語に対応する前記ＨＤＬテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換するステップとを備えている。

【０００９】本発明によるＨＤＬ仕様変換制御プログラムを記録した記録媒体は、コンピュータに、入力されたハードウェア記述言語を変換したいハードウェア記述言語に変換させるためのＨＤＬ仕様変換制御プログラムを記録した記録媒体であって、前記ＨＤＬ仕様変換制御プログラムは前記コンピュータに、複数種類のハードウェア記述言語各々の構文規則及び予約語と構文要素との位置関係が予め定義されたＨＤＬテンプレートを記憶するテンプレートライブラリを参照して前記入力されたハードウェア記述言語から前記構文要素を取出させ、その取出させた構文要素を格納させ、格納させた前記構文要素を基に前記変換したいハードウェア記述言語に対応する前記ＨＤＬテンプレートを参照して前記入力されたハードウェア記述言語を前記変換したいハードウェア記述言語に変換させている。

【００１０】すなわち、本発明のＨＤＬ仕様変換システムは、様々な種類がありかつ夫々構文規則が異なるＨＤＬにおいて、異種類のＨＤＬであっても同じ意味を持つ構文を構成する要素が同じである点に着目し、ＨＤＬの構文要素に着目したデータフィールドをもつ設計回路情報を格納するデータベースと、ある特定種類のＨＤＬの構文規則及びその構文の要素とデータベースとの間の対応規則を書いたテンプレートライブラリと、そのライブラリを参照してＨＤＬからデータベースへの変換を行う第１の変換手段と、同じくライブラリを参照してデータベースからＨＤＬへの変換を行う第２の変換手段とを有している。

【００１１】ある種類のＨＤＬはその種類にあったテンプレートライブラリを参照して第１の変換手段が構文毎にその要素を取出し、その要素の情報をデータベースへ格納する。

【００１２】このデータベースの内容を基に第２の変換手段が別の種類用のライブラリを参照して構文規則に従って要素を再構成し、その再構成結果を出力する。これによって、結果的にある種類のＨＤＬが別種類のＨＤＬに変換されるので、人手を使わず、またＨＤＬ種類個別にプログラムを用意することなく、容易に多種のＨＤＬ同士を変換することが可能となる。

【００１３】

【発明の実施の形態】次に、本発明の一実施例について図面を参照して説明する。図１は本発明の一実施例によるＨＤＬ仕様変換システムの構成を示すブロック図である。図において、本発明の一実施例によるＨＤＬ仕様変換システムはＨＤＬ入力（読込）装置１と、データ処理装置２と、記憶装置３と、ＨＤＬ出力装置４とから構成されている。

【００１４】ＨＤＬ入力装置１はＨＤＬを入力するものであり、キーボードや他の記憶媒体からＨＤＬを入力する入力機器からなる。データ処理装置２は構文要素取出し手段２１とＨＤＬ構成手段２２とを持ち、図示せぬ制御メモリのプログラムを基に動作する。尚、制御メモリとしてはＲＯＭ（リードオンリメモリ）やフロッピディスク等が使用可能である。

【００１５】記憶装置３は予め設定された複数種類のＨＤＬテンプレートを記憶するＨＤＬテンプレート記憶部３１と、構文要素取出し手段２１で取出された構成要素を記憶する構成要素記憶部３２（データベース）とからなる。ＨＤＬ出力装置４はＨＤＬ構成手段２２で構成されたＨＤＬをディスプレイ装置（図示せず）や他の記憶媒体に出力する出力機器である。

【００１６】ここで、ＨＤＬテンプレート記憶部３１に予め記憶されたＨＤＬテンプレートには各種ＨＤＬの構文規則及び予約語と構文要素との位置関係が予め定義されている。

【００１７】ＨＤＬ入力装置１に読込まれたＨＤＬ（例えば、種類Ａ）はデータ処理装置２の構文要素取出し手段２１に渡される。構文要素取出し手段２１はＨＤＬ入力装置１から読込まれたＨＤＬの種類に合ったＨＤＬテンプレート（種類Ａ）をＨＤＬテンプレート記憶部３１から呼出し、そのＨＤＬテンプレートを使用してＨＤＬ（種類Ａ）から各構文要素を取出す。構文要素取出し手段２１によって取出された各構文要素は構成要素記憶部３２に格納される。

【００１８】構成要素記憶部３２に格納された各構文要素はＨＤＬ構成手段２２に渡される。ＨＤＬ構成手段２２は変換したいＨＤＬの種類（例えば、種類Ｂ）に合ったＨＤＬテンプレート（種類Ｂ）をＨＤＬテンプレート記憶部３１から呼出し、そのＨＤＬテンプレートを参照してＨＤＬ（種類Ｂ）の予約語と構文要素とを合わせてＨＤＬ（種類Ｂ）に再構成する。ＨＤＬ構成手段２２で再構成されたＨＤＬ（種類Ｂ）はＨＤＬ出力装置４によ

ってディスプレイ装置や他の記憶媒体に出力される。

【００１９】図２は本発明の一実施例によるＨＤＬ仕様変換システムの動作を示すフローチャートであり、図３は図２のデータベースのフォーマットの一例を示す図であり、図４は本発明の一実施例によるＨＤＬ→データベース変換の一例を示す図であり、図５及び図６は本発明の一実施例によるデータベース→ＨＤＬ変換の一例を示す図である。

【００２０】図３において、このデータベースのレコード（行）はＨＤＬの構文用途（意味）で分けられ、フィールドは”；”で区切られ、その構文の要素が格納されている。

【００２１】ここで、ｎｕｍは識別番号を、ｗｏｒｄは構文を表すキーワード（例えば、レジスタの場合、ｒｅｇｉｓｔｅｒ）を、＄１はクロック信号名を、＄２はクロックエッジ（立上りまたは立下り）を、＄３は出力信号名ービット範囲を、＄４は入力信号名ービット範囲を夫々示している。但し、＄１以降はキーワードによって意味が変わる。

【００２２】テンプレートライブラリでは構文種類と、その構文種類がそのＨＤＬ種類でどのような構文規則で記述されているかが定義されている。構文規則とはそのＨＤＬ種類での予約語と構文要素とそれらの位置関係とを示す。テンプレートライブラリはＨＤＬを読込む場合、あるいはＨＤＬを出力する場合に同じものを使用する。例えば、ＨＤＬ（種類Ａ）を読込む場合と、ＨＤＬ（種類Ａ）で出力する場合とは同じＨＤＬ（種類Ａ）用のテンプレートライブラリを使用する。

【００２３】構文要素を取出す場合には予約語との位置関係をテンプレートで見て、定義されている位置にあるものを構文要素として抽出する。ＨＤＬに出力する場合には構文要素を定義されている位置に入れ、予約語と合わせて全て出力する。ＨＤＬ（種類Ａ）の例［図４の（１）参照］でいえば、ｒｅｇｉｓｔｅｒ／／の後に書かれているのがＨＤＬ（種類Ａ）でのレジスタ構文の規則になる。

【００２４】”ＲＥＧ”、”ＩＦ”、”ＴＨＥＮ”はＨＤＬ（種類Ａ）の予約語であり、”ＲＥＧ”の次の位置にあるものが「＄３＝レジスタ構文の出力信号名」、”ＩＦ”の次の位置にあるものが「＄１＝レジスタ構文のクロック信号名」という構文要素になる。

【００２５】テンプレートライブラリはＨＤＬ間で同意の構文要素をデータベース中の同じレコードと対応するようにして作成される。ＨＤＬ（種類Ａ）で書かれた論理はＨＤＬ（種類Ａ）用のテンプレートライブラリ３１ａを参照して構文要素取出し手段２１によって読込まれ、構文要素が取出される（図２ステップＳ１）。取出された構文要素はデータベース５（構成要素記憶部３２）に出力される（図２ステップＳ２）［出力例は図４の（２）参照］。

【００２６】図４の（２）はＨＤＬ（種類Ａ）でのレジスタを表す構文であり、ＨＤＬ（種類Ａ）のテンプレートライブラリ［図３の（１）参照］のレジスタ構文部中の「＄１」の箇所にあたる要素”ＣＬＫ”がクロック信号名として取出される。同様に、「クロックエッジ（＄２）」として「立上がり」、「出力信号名（＄３）」として”ＰＳ（０）～（３）”、「入力信号名（＄４）」として”Ｄ（０）～（３）”がテンプレートライブラリを参照して要素として取出され、データベースに出力される［図４の（３）参照］。

【００２７】データベースに出力された各要素はデータベースの読込みによって同意のレコードから取出され、別種類のテンプレートライブラリ３１ｂを参照してＨＤＬ構成手段２２によってライブラリ中の記述仕様に基づいて出力される（図２ステップＳ３）。

【００２８】データベース［図５の（１）参照］でレジスタ構文の＄１のフィールド”ＣＬＫ”はＨＤＬ（種類Ｂ）のテンプレートライブラリ［図５の（２）参照］を参照した場合、”レジスタ構文”＄１の箇所に出力される。＄２、＄３、＄４も夫々ＨＤＬ（種類Ｂ）のテンプレートライブラリで定義されている箇所に出力される。その結果、入力されたＨＤＬ（種類Ａ）はＨＤＬ（種類Ｂ）に変換されてＨＤＬ出力装置４から出力される（図２ステップＳ４）［図５の（３）参照］。

【００２９】同様に、ＨＤＬ（種類Ｃ）のテンプレートライブラリを使用すれば、ＨＤＬ（種類Ｃ）での”レジスタ構文”で出力される（図６参照）。

【００３０】このように、構文要素取出し手段２１が入力されたＨＤＬ（種類Ａ）に対応するＨＤＬ（種類Ａ）テンプレートライブラリ３１ａを参照して構文毎にその要素を取出し、その要素の情報を構成要素記憶部３２（データベース５）に格納し、ＨＤＬ構成手段２２が構成要素記憶部３２（データベース５）に格納された構成要素を基に、ＨＤＬ（種類Ｂ）テンプレートライブラリ３１ｂを参照して構文規則に従って要素を再構成し、その再構成結果をＨＤＬ出力装置４から出力することによって、人手を使わず、またＨＤＬ種類個別にプログラムを用意することなく、容易に多種のＨＤＬ同士を変換することができる。また、ＨＤＬだけでなく、ゲートレベルのネットリストでも同様に多種類間の変換を容易に行うことができる。

【００３１】

【発明の効果】以上説明したように本発明によれば、複数種類のハードウェア記述言語の構文規則及び予約語と構文要素との位置関係が予め定義されたＨＤＬテンプレートを記憶するテンプレートライブラリを参照して入力されたハードウェア記述言語から構文要素を取出し、その取出された構文要素を格納し、格納された構文要素を基に変換したいハードウェア記述言語に対応するＨＤＬテンプレートを参照して入力されたハードウェア記述言

語を変換したいハードウェア記述言語に変換することによって、人手を使わず、またＨＤＬ種類個別にプログラムを用意することなく、容易に多種のＨＤＬ同士を変換することができるという効果がある。

【図面の簡単な説明】

【図１】本発明の一実施例によるＨＤＬ仕様変換システムの構成を示すブロック図である。

【図２】本発明の一実施例によるＨＤＬ仕様変換システムの動作を示すフローチャートである。

【図３】図２のデータベースのフォーマットの一例を示す図である。

【図４】本発明の一実施例によるＨＤＬ→データベース変換の一例を示す図である。

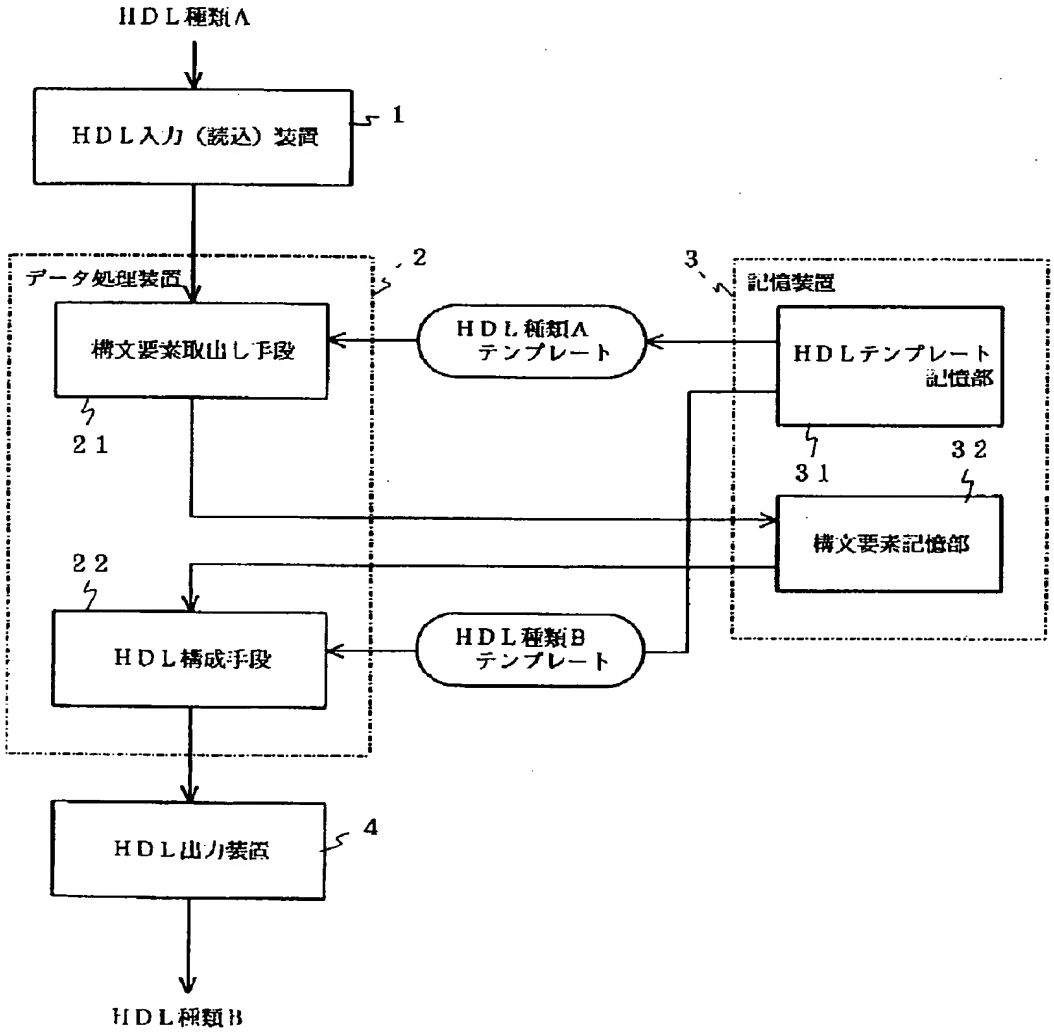
【図５】本発明の一実施例によるデータベース→ＨＤＬ変換の一例を示す図である。

【図６】本発明の一実施例によるデータベース→ＨＤＬ変換の一例を示す図である。

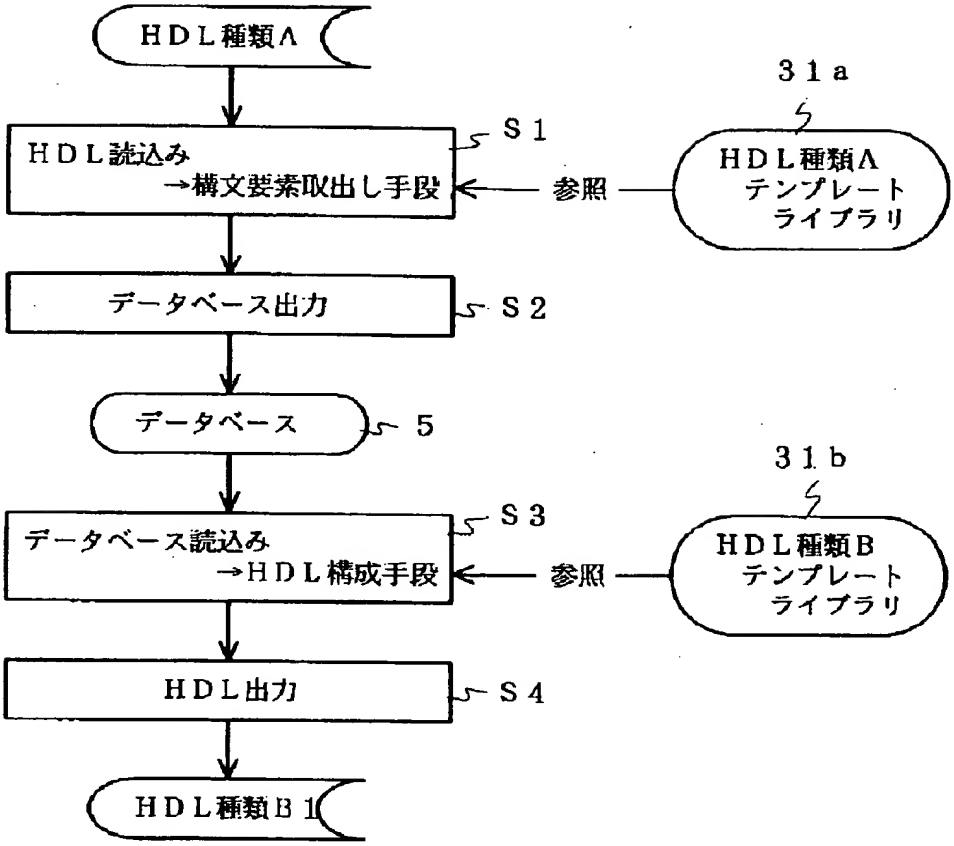
【符号の説明】

１　ＨＤＬ入力（読込）装置
２　データ処理装置
３　記憶装置
４　ＨＤＬ出力装置
５　データベース
２１　構文要素取出し手段
２２　ＨＤＬ構成手段
３１　ＨＤＬテンプレート記憶部
３１ａ　ＨＤＬ種類Ａテンプレートライブラリ
３１ｂ　ＨＤＬ種類Ｂテンプレートライブラリ
３２　構成要素記憶部

【図１】



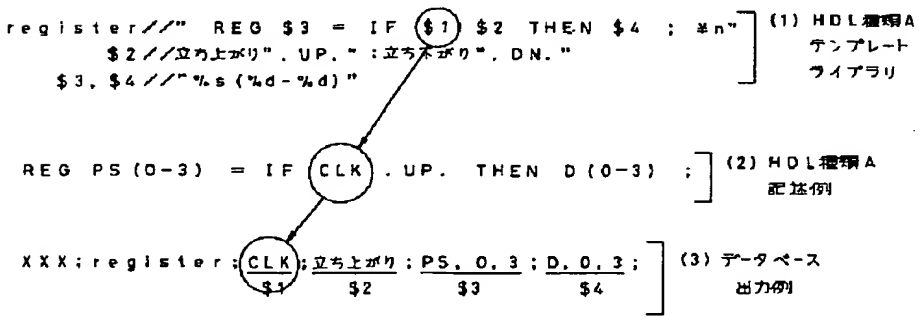
【図２】



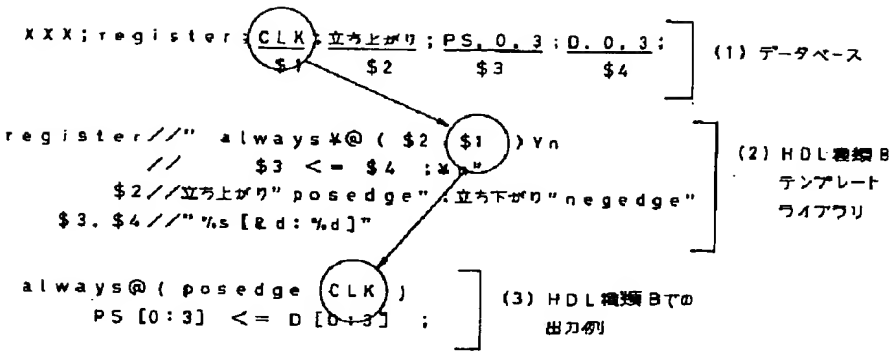
【図３】

num;word;$1;$2;$3;$4
　num ： 識別番号
　word ： 構文を表すキーワード（レジスタの場合 register）
　$1： クロック信号名
　$2： クロックエッジ（立ち上がり or 立ち下がり）
　$3： 出力信号名 ビット範囲
　$4： 入力信号名 ビット範囲；　$1からは、キーワードにより意味が変わる。

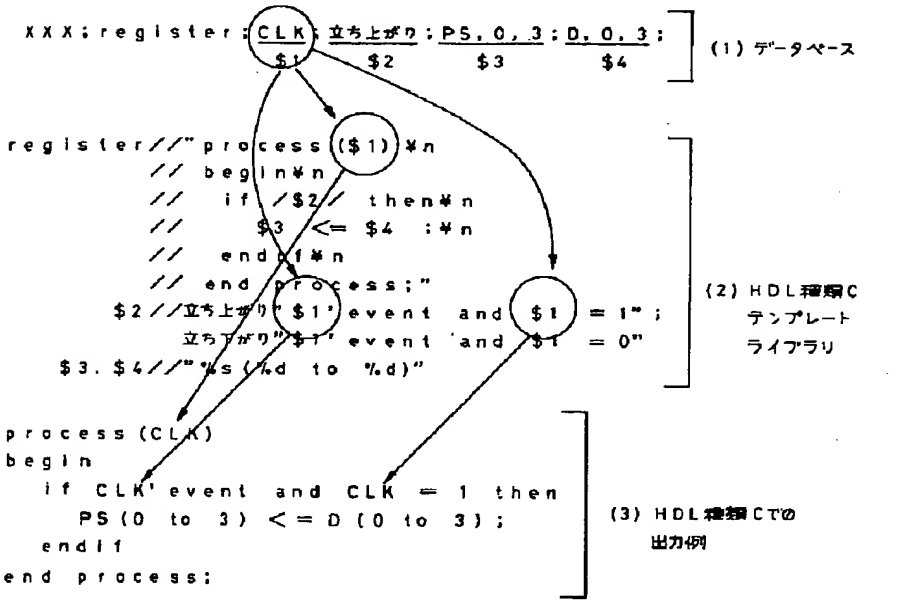
【図４】



【図５】


XXX;register;CLK;立ち上がり;PS,0,3;D,0,3;
$1 $2 $3 $4
(1) データベース
register//" always¥@ ($2 $1)¥n
// $3 <= $4 ;¥n"
$2//立ち上がり" posedge";立ち下がり" negedge"
$3, $4//"%s [&d:%d]"
(2) HDL種類B テンプレート ライブラリ
always@(posedge CLK)
PS [0:3] <= D [0:3] ;
(3) HDL種類Bでの出力例

【図６】


XXX;register;CLK;立ち上がり;PS,0,3;D,0,3;
$1 $2 $3 $4
(1) データベース
register//"process ($1) ¥n
// begin¥n
// if /$2// then¥n
// $3 <= $4 ;¥n
// end if¥n
// end process;"
$2//立ち上がり"$1'event and $1 = 1";
立ち下がり"$1'event and $1 = 0"
$3, $4//"%s (%d to %d)"
(2) HDL種類C テンプレート ライブラリ
process (CLK)
begin
if CLK'event and CLK = 1 then
PS(0 to 3) <= D(0 to 3);
end if
end process;
(3) HDL種類Cでの出力例